

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
Ingenieros de Sistemas**

**TEMA:
EVALUACIÓN DEL PROTOTIPO PIZARRA DIGITAL INTERACTIVA,
PARA EL PLANTEAMIENTO DE MEJORAS FUNCIONALES Y
OPTIMIZACIÓN.**

**AUTORES:
DARWIN JEFFERSON AGUAYO MACAS
ALEXIS FABRICIO ZAPATA CISNEROS**

**TUTOR:
FRANKLIN EDMUNDO HURTADO LARREA**

Quito, agosto de 2019

CESIÓN DE DERECHOS DE AUTOR

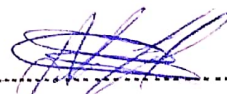
Nosotros, Darwin Jefferson Aguayo Macas y Alexis Fabricio Zapata Cisneros, con documento de identificación N.º 1723508626 y N.º 1724602618, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: "EVALUACIÓN DEL PROTOTIPO PIZARRA DIGITAL INTERACTIVA, PARA EL PLANTEAMIENTO DE MEJORAS FUNCIONALES Y OPTIMIZACIÓN", mismo que ha sido desarrollado para optar por el título de: INGENIEROS DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.



**DARWIN JEFFERSON
AGUAYO MACAS**

C.I.: 1723508626



**ALEXIS FABRICIO
ZAPATA CISNEROS**

C.I.: 1724602618

Quito, agosto de 2019

DECLARATORIA DE COAUTORÍA DEL TUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado el Artículo Académico, con el tema: "EVALUACIÓN DEL PROTOTIPO PIZARRA DIGITAL INTERACTIVA, PARA EL PLANTEAMIENTO DE MEJORAS FUNCIONALES Y OPTIMIZACIÓN" realizado por Darwin Jefferson Aguayo Macas con C.I.: 1723508626 y Alexis Fabricio Zapata Cisneros con C.I.: 1724602618, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerados como trabajo final de titulación.

Quito, agosto de 2019



Franklin Edmundo Hurtado Larrea

C.I.: 1713382016

DEDICATORIA

Quiero dedicar este trabajo principalmente a mis padres José Aguayo y Carmita Macas; han sido la piedra angular de mi vida y la culminación de mi carrera es fruto de todos sus esfuerzos por darme lo mejor, sembrando en mí, la perseverancia, valores y su amor incondicional. Ha sido un objetivo que hemos soñado juntos y por fin lo hemos logrado.

A mi querida familia entera también, mis hermanos por ser parte de este largo camino, mis abuelos José Facundo y Luz Ercilia (q.e.p.d), que sus palabras de aliento y los consejos perduraron y perdurarán en mi corazón por siempre. A mi padrino, el Dr. Carlos Aguay y a mi tía Alicia Judith que han sido como mis segundos padres, han formado de mí una persona con muchos sueños y ganas de seguir triunfando.

A todas las personas que alguna vez formaron parte de este largo camino, que brindaron parte de su vida, de su tiempo y cada paso vivido me ha hecho crecer como persona y les guardo con gratitud en mi corazón.

Darwin Aguayo

AGRADECIMIENTO

“¿Por qué esta magnífica tecnología científica, que ahorra trabajo y nos hace la vida más fácil, nos aporta tan poca felicidad? La respuesta es está, simplemente: porque aún no hemos aprendido a usarla con tino.”

-Albert Einstein

Un agradecimiento especial a nuestro tutor el Ingeniero Franklin Hurtado quien depositó su confianza para poder culminar este trabajo de titulación, además por contribuir a nuestra formación académica y personal.

A la Universidad Politécnica Salesiana y a sus maestros por compartir su tiempo e impartir sus conocimientos, sus experiencias, el cariño por esta carrera haciendo que nuestros objetivos de vida sean consumados con éxito y proveernos de las herramientas suficientes para labrar nuestro destino.

A mi compañero y amigo Alexis quien por fin hemos logrado con tanto sacrificio culminar esta meta, donde hemos podido aprender más y apoyarnos también durante todo este tiempo.

A mis amigos con quien he podido compartir momentos agradables dentro y fuera de la Universidad quien me han permitido entrar en sus vidas, aprender de ellos, donde se han permitido escucharme y entenderme.

A todos quienes en mi pensamiento los tengo presentes. ¡Gracias!

Darwin Aguayo

DEDICATORIA

A mi madre Carmen Zapata y mi abuelita Luz Cisneros por su amor, su apoyo y la fuerza que se me brindó a través de estos años para culminar uno de tantos objetivos de vida; ante todo son mis madres y sus enseñanzas han llevado a su hijo a lograr un sueño, para ustedes con mucho amor madres.

A mis tíos Francisco Zapata y Víctor Zapata, a mis tías Margaret Zapata y Elena Zapata por su apoyo incondicional, por inculcar en mí la perseverancia para alcanzar las metas. A mi familia por estar presentes con su cariño y constancia a lo largo de este arduo camino.

A los amigos y compañeros que estuvieron presentes todos estos años, siempre serán parte de buenos recuerdos y gratas experiencias. Por todo el tiempo compartido y las enseñanzas que nos dejó la carrera, ustedes son parte de este logro.

Alexis Zapata

AGRADECIMIENTO

Ing. Franklin Hurtado gracias por depositar su confianza en nosotros y por ser parte de este camino, sus enseñanzas nos ayudaron a conciliar nuestros objetivos. Este proyecto es un logro de su entusiasmo por enseñar e inculcar la pasión por la carrera.

A todos los maestros por su enseñanzas y experiencias compartidas en las aulas, por demostrarnos que la carrera es compleja pero bella. Son y serán la base fundamental de nuestros logros académicos y profesionales.

Amigo Darwin fue un camino difícil, pero lo logramos, gracias por el apoyo a lo largo de estos años, estoy seguro que la vida aun nos tiene preparado grandes proyectos.

A los amigos y amigas que conocimos en las aulas y fuera de ellas, gracias por ser parte de este triunfo, siempre estarán presentes en mis mejores recuerdos.

Alexis Zapata

EVALUACIÓN DEL PROTOTIPO PIZARRA DIGITAL INTERACTIVA, PARA EL PLANTEAMIENTO DE MEJORAS FUNCIONALES Y OPTIMIZACIÓN

Darwin Jefferson Aguayo Macas, Alexis Fabricio Zapata Cisneros,
Franklin Edmundo Hurtado Larrea
Universidad Politécnica Salesiana Ecuador
Quito, Ecuador

E-mail: daguayo@est.ups.edu.ec

E-mail: azapataci@est.ups.edu.ec

E-mail: fhurtado@ups.edu.ec

Abstract — This article describes the software re-engineering process applied to the first prototype project: Interactive Digital Whiteboard; By adapting this model, its current state is evaluated and important features of its constitution are identified, such as architecture, functionalities and the different components that are involved in the mobile application and the server application. From this process, relevant information is obtained that allows identifying optimizations and new functionalities.

Later this analysis as part of the optimization is implemented a new architecture allows the bifurcation of services and reduces the load on the server, in this way it gives better response times and a new communication method is used between the mobile application and The server, which improves the interaction of users and allows to implement new functional improvements, such as a second prototype of the Interactive Digital Whiteboard (PDI).

In section II, software reengineering is conceptualized as a technique to improve or optimize an existing application. In section III, we talk about the reengineering process carried out for this research project. Section IV presents the optimizations implemented in the second PDI prototype. Section V reflects the results obtained, added and suggested functionalities, as well as the resulting application. Finally, sections VI and VII present the discussion of the research carried out and conclusions of the present project.

Key words — Reengineering, software optimization, software maintenance.

Resumen — El presente artículo describe el proceso de reingeniería de software aplicado al primer

proyecto del prototipo: Pizarra Digital Interactiva; mediante la adaptación de este modelo se evalúa su estado actual y se identifica características importantes de su constitución como su arquitectura, funcionalidades y los diferentes componentes que están involucrados entre la aplicación móvil y la aplicación servidor. A partir de este proceso se obtiene información relevante que permite identificar futuras optimizaciones y nuevas funcionalidades.

Posterior a este análisis, como parte de la optimización se implementa una nueva arquitectura que permite la bifurcación de servicios y reduce la carga sobre el servidor, de esta manera se obtienen mejores tiempos de respuesta y se emplea un nuevo método de comunicación entre la aplicación móvil y el servidor, lo cual mejora la interacción de los usuarios y permite implementar nuevas mejoras funcionales, dando como resultado la segunda versión del prototipo Pizarra Digital Interactiva (PDI).

En el apartado II se conceptualiza la reingeniería de software como una técnica para mejorar u optimizar una aplicación existente. En el apartado III se habla sobre el proceso de reingeniería llevado a cabo para este proyecto de investigación. En el apartado IV se presenta las optimizaciones implementadas en el segundo prototipo de la PDI. El apartado V refleja los resultados obtenidos, funcionalidades agregadas y sugeridas, así como la aplicación resultante. Finalmente, el apartado VI y VII presenta la discusión de la investigación realizada y conclusiones del presente proyecto.

Palabras Clave — Reingeniería, optimización de software, mantenimiento de software.

1 Introducción

En la última década la tecnología se ha mantenido en un punto clave de innovación y evolución. La industria tecnológica brinda nuevas mejoras en sus productos, buscando de esta manera predominar sobre los productos de la competencia, lo que resulta sin duda, en la evolución constante de hardware y software. Esto permite la mejora continua en el desarrollo de aplicaciones móviles, presentando nuevas innovaciones en entretenimiento, videojuegos, educación, salud, comunicación, entre otros [1].

Las actualizaciones permiten mejorar la seguridad, productividad, rendimiento, interoperabilidad entre otras propiedades y características propias del software, presentando así, oportunidades para que las aplicaciones móviles se vuelvan más competitivas y atractivas hacia el usuario. [2]

Aunque existen factores inhibidores para implementar actualizaciones como: la resistencia al cambio, costes o interdependencia con otros sistemas; dando como resultado, a corto o largo plazo la posible inutilización u obsolescencia del software, se considera de vital importancia la implementación de las mejoras presentadas por los desarrolladores, dado que ayudan o permiten el perfeccionamiento y a la optimización de los servicios ofrecidos por estas aplicaciones. [2]

El soporte inexistente o mínimo de los sistemas informáticos provoca el estancamiento de los mismos, llevándolos a un problema de desactualización y provocando un funcionamiento ineficiente o limitado al que presentaban iniciado su producción. [3]

Actualmente el prototipo de la PDI (Pizarra Digital Interactiva) se encuentra sin soporte que posibilite la mejora del rendimiento o funcionamiento. Al mismo tiempo no se han tomado en cuenta las actualizaciones, ni el progreso tecnológico que ha existido en los últimos años (desde la culminación del primer prototipo) [4] [5] que puedan ayudar a mantener el servicio activo.

Ciertamente dentro del proceso de desarrollo de software se contempla el mantenimiento que considera la implementación de actualizaciones y soporte para estos productos, el resultado de este proceso recae en la estabilidad de las aplicaciones, proporcionando fiabilidad en su uso diario. [3]

Se plantea entonces la evaluación del prototipo de la PDI que ayude a entender su

constitución y los componentes por los cuales está conformado, de esta manera mejorar su estructura y solventar los problemas existentes en su primera versión. A través del modelo de reingeniería de software [6] [7] se busca conseguir nuevamente la estabilización de la aplicación móvil y el servidor. Finalmente, proyectarse hacia la optimización y buscar mejoras funcionales, llevando a esta aplicación a un nuevo nivel. La experiencia que se intenta brindar al usuario final debe ser más amigable e impecable tanto en eficiencia como funcionalidad.

2 Reingeniería y optimización del software

El proceso evolutivo del software parte del cambio y la adaptación, siendo fundamental la mejora continua como parte este proceso. La reingeniería de software presenta un modelo flexible que permite adaptarse a un caso específico de estudio para mejorar su estructura y entendimiento, dentro de este modelo la optimización es un factor¹ resultante que permite elevar la calidad del software.

La reingeniería de software es el conjunto de actividades que permite la transformación de un sistema mediante la revisión, el estudio y el análisis, logrando su comprensión y estableciendo mejoras totales o parciales a partir de un sistema ya existente. [6] [7]

Según Roger S. Pressman, el conjunto de actividades que comprenden el modelo del proceso de reingeniería de software son las siguientes:

- **Reestructuración de documentos:** Permite la creación, actualización y redocumentación de la información existente de un sistema. [6] [7]
- **Ingeniería inversa:** Es el análisis y extracción de información de un sistema, con el propósito de establecer una representación de su diseño. [6] [7]
- **Reestructuración de código:** Es el análisis detallado, manual o automático del código fuente para eliminar anomalías o redundancias dentro de su estructura. [6] [7]
- **Reestructuración de datos:** Es la redefinición de los esquemas de datos, así como de su estructura para corregir errores o datos redundantes entre otros. [6] [7]

¹ Elemento, circunstancia, influencia, que contribuye a producir un resultado.

- **Ingeniería hacia adelante:** Es la implementación de un nuevo diseño a partir de funcionalidades y tecnologías para mejorar la eficiencia de un sistema. [6] [7]

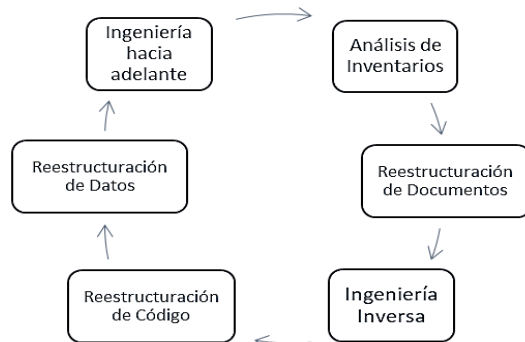


Figura 1. Modelo del proceso de Reingeniería de software

Cada una de estas actividades definen el procedimiento para establecer el nuevo diseño y/o estructura a implementarse de un sistema previamente analizado. [6] [7]

La reingeniería plantea una solución a este dilema presente en el software, cuando el mantenimiento o soporte estuvo presente de forma limitada o nula. Posterior a este proceso se debe establecer también pruebas que permitan valorar el software.

En el estándar ISO 9126 se establece el modelo de calidad de software (interno y externo), así como su modelo de calidad de uso [8], estos se componen de características y subcaracterísticas que permiten, a partir de métricas evaluar a un software. [9] [10]

En el modelo de calidad interno y externo se pueden considerar las siguientes características:

- **Funcionalidad:** Es el conjunto de atributos que hacen referencia a las funciones y propiedades que satisfacen las necesidades de un sistema. [11]
- **Eficiencia:** Se relaciona al nivel de desempeño y la cantidad de recursos necesarios para el funcionamiento óptimo del software. [11]
- **Efectividad:** Es la capacidad del software para ayudar al usuario a lograr sus objetivos con exactitud. [11]
- **Satisfacción:** El nivel de conformidad y aceptación que proporciona un software respecto a las necesidades y funcionalidades establecidas. [11]

Se considera la optimización como el planteamiento e implementación de mejoras a partir del enfoque de características de calidad

(eficiencia, funcionalidad, usabilidad entre otros) para mejorar los atributos de un sistema de software y lograr su perfeccionamiento. [9] [12] [10]

3 Proceso de reingeniería aplicado al primer prototipo de la aplicación pizarra digital interactiva

El modelo de reingeniería de software presentado por Roger S. Pressman no es necesariamente un proceso cíclico o estricto, las actividades dentro del modelo pueden aplicarse dependiendo de las necesidades del caso de estudio. [6]

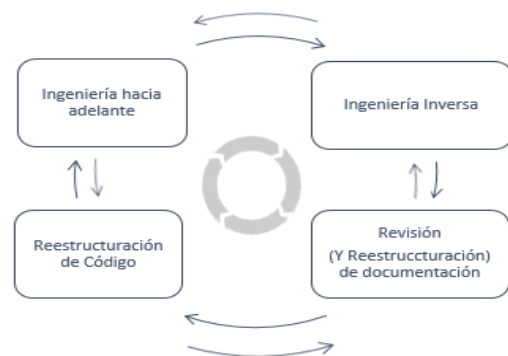


Figura 2. Modelo del proceso de Reingeniería de software aplicado al primer prototipo de la PDI

El proceso de reingeniería parte del entendimiento de la PDI (aplicación móvil y servidor), por esta razón se inicializa con la recolección de documentación previamente creada en su primer prototipo que explique su funcionalidad, estructura y los componentes que lo conforman.

En el primer prototipo la documentación fue limitada; no se obtuvo mayor representación de su constitución lo cual dificultaba la comprensión del sistema.

Inicialmente se puede afirmar que la aplicación Pizarra Digital Interactiva se constituye como un sistema que comprende dos partes:

- **Aplicación móvil:** Desarrollada bajo el lenguaje JAVA [13] y Android [14], está constituida por funcionalidades como registro de usuario, inicio de sesión, consulta de contactos, agregación de contactos, videollamada y lienzo que permite la interacción de los usuarios en tiempo real.
- **Aplicación Servidor:** Desarrollado bajo el lenguaje PHP y la estructura

cliente/servidor [15], ofrece servicios web los cuales son precisados por la aplicación móvil para su correcto funcionamiento, además de la comunicación con la base de datos (MySQL) [16] y gestión del archivo lienzo el cual, es un archivo plano que permite la escritura y lectura de los trazos.

3.1 Ingeniería Inversa

El proceso de ingeniería inversa se lo aplicó a los dos componentes del prototipo PDI a través de los siguientes pasos:

A. *Aplicación Servidor*

Pasos realizados en el proceso de ingeniería inversa:

1. Identificar los servicios proporcionados por el servidor.
2. Identificar la conexión de los servicios con las funcionalidades de la aplicación móvil.
3. Diseñar grafos y diagramas UML para relacionar las funciones del servidor y aplicación móvil [17].
4. Vincular las clases, métodos y cómo interactúan dentro del servidor con las respectivas funciones.
5. Crear o desarrollar un entorno que permita testear cada uno de los servicios.
6. Detectar los errores, bugs o transferencias inconclusas en la petición y respuesta de datos.

B. *Aplicación Móvil*

Pasos realizados en el proceso de ingeniería inversa:

1. Identificar las funcionalidades proporcionadas por la aplicación móvil.
2. Identificar la conexión de las funcionalidades con los servicios de la aplicación servidor.
3. Diseñar grafos y diagramas UML para relacionar las funciones de la aplicación móvil y servidor.
4. Vincular las clases, métodos, API's (SINCH SERVICES) [18] comprobando la interacción dentro de la aplicación móvil con las respectivas funciones.
5. Testear las funcionalidades mediante el IDE (Interface Development Environment) [14] a través de un emulador Android.
6. Detectar los errores, bugs o transferencias inconclusas en la petición y respuesta de datos.

3.2 Revisión (y reestructuración) de documentos

El enfoque aplicado en esta actividad fue complementar la documentación que ayude a la comprensión del primer prototipo mediante artefactos que comuniquen su estructura y demás componentes que lo constituyen [19].

3.3 Reestructuración de Código

Proceso de reestructuración de código llevado a cabo para los componentes del prototipo PDI

A. *Aplicación Servidor*

Pasos realizados en la reestructuración del código dentro del servidor:

1. Reestructurar el código como resultado de la actualización de PHP 5 versión 5.3.8 a la versión 7.3.1 [20]
2. Adecuar la estructura de consultas a la base de datos mediante PDO aceptadas por MySQL 4.8.4 [20] [21]
3. Ajustar las clases a una estructura similar a la del patrón estructural ADAPTER. [22]

B. *Aplicación Móvil*

Pasos realizados en la reestructuración del código dentro de la aplicación PDI:

1. Reestructurar el código como resultado de la actualización de versión de Android Studio 3.3
2. Adaptar la aplicación a la versión de Android Pie SDK 28
3. Configurar librerías y dependencias basadas en Android Pie [23]
4. Agregar comentarios en bloques de código fuente para mejorar su comprensión y ayudar en futuros mantenimientos.

3.4 Ingeniería hacia adelante

Proceso de ingeniería hacia adelante llevado a cabo para los componentes del prototipo PDI.

A. *Aplicación Servidor*

Pasos realizados en el proceso de ingeniería hacia adelante:

1. Separar los métodos en dos clases (Contacto y Usuario) basado en los servicios requeridos por las funcionalidades de la aplicación móvil.
2. Implementar la bifurcación de servicios respecto a las funcionalidades específicas para la comunicación del lienzo y del usuario. [24]

3. Implementar una nueva arquitectura basada en la repartición de carga. [25]
4. Eliminar código y clases redundantes o sin uso específico dentro de la reestructuración.

B. Aplicación Móvil

Pasos realizados en el proceso de ingeniería hacia delante implementado en la aplicación Pizarra Digital Interactiva:

1. Actualizar el SDK SINCH [26] para el servicio de video llamada compatible con el sistema Android 6.0 o superior.
2. Adaptar nuevas interfaces basadas en librerías de Android Pie.
3. Implementar método para la creación de archivos de interacción por cada usuario registrado para el uso del lienzo.
4. Implementar el uso de las interfaces de servicio de videollamada como complemento para establecer la lógica de comunicación mediante el uso de archivos de interacción dentro del servidor lienzo.
5. Implementar un método basado en un ciclo repetitivo mediante un hilo que permite iniciar y detener el proceso de lectura de los trazos. [27]

4 Optimizaciones

En base al proceso de reingeniería se obtuvo información fundamental la cual permitió conocer el estado de la aplicación. A continuación, se describe cada una de las optimizaciones implementadas dentro del segundo prototipo de la PDI.

a. Arquitectura utilizando la distribución de carga

La arquitectura implementada en el primer prototipo de la aplicación PDI plantea un modelo basado en la arquitectura cliente/servidor [28]. La propuesta actual presenta una solución a la limitante en las peticiones hacia el servidor y está basada en la distribución de carga, a través de ello se dividen los servicios y la carga se ve repartida hacia dos servidores [25], de esta forma la sincronización de los servicios que se emplean dentro de la aplicación móvil reduce notablemente los tiempos de respuesta, es decir, mejora la experiencia con el usuario.

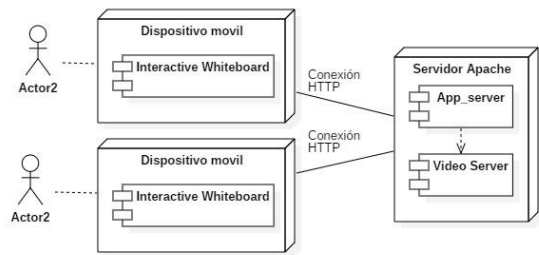


Figura 3. Arquitectura del Primer prototipo de la PDI

La arquitectura planteada en la figura 4 divide los servicios de la arquitectura inicial (figura 3) en dos segmentos [24], la primera gestiona las principales funcionalidades que ofrece la aplicación móvil excluyendo la funcionalidad Gestionar Pizarra.

Esta funcionalidad en específico es gestionada por el servidor lienzo, el cual gestiona la creación del archivo plano para escritura y lectura de los trazos realizados dentro de la pizarra.

La aplicación móvil mantiene su arquitectura original para la gestión de información correspondiente a cada usuario. Se conserva el formato JSON (JavaScript Object Notation) [29] para el intercambio de datos desde la aplicación móvil al servidor, pero con la reestructuración implementada en la aplicación servidor.

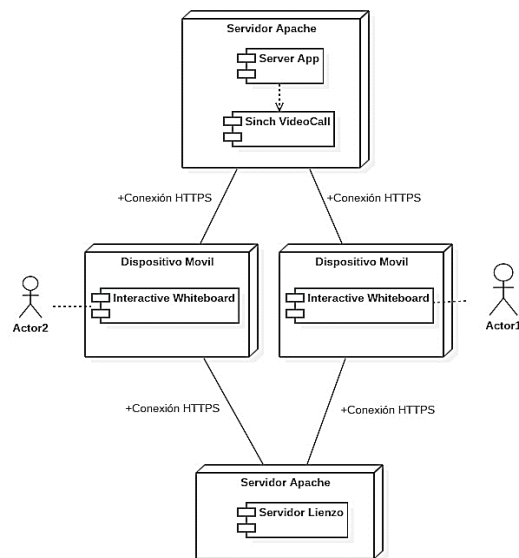


Figura 4. Arquitectura del Segundo prototipo de la PDI

El servicio de videollamada de SINCH [26] continúa siendo parte del segundo prototipo de la PDI, ya que su estructura se adapta a las necesidades de la actual arquitectura.

La arquitectura implementada en el segundo prototipo de la PDI plantea un reto, al dividir los servicios se debe establecer un medio de

comunicación que permita conocer a la aplicación móvil, cuál es el archivo que debe usar para empezar la interacción entre los dos usuarios. En el siguiente apartado se expone la solución que se implementó para establecer este medio de comunicación.

b. Optimización en la lógica de comunicación para el servidor lienzo

En el primer prototipo de la aplicación móvil PDI se creaba un archivo de texto, el cual permitía la escritura, así como la lectura que fuese el resultado de la interacción dentro del lienzo, este permitía que los usuarios visualicen los trazos plasmados en tiempo real.

El inconveniente que se presentaba en el primer prototipo es que este archivo dificultaba las interacciones de más de dos usuarios al mismo tiempo.

Se hace uso de las interfaces de servicio de videollamada como complemento para establecer la lógica de comunicación, usando también archivos de interacción dentro del servidor lienzo. El siguiente diagrama de flujo representa el proceso llevado a cabo en esta solución.

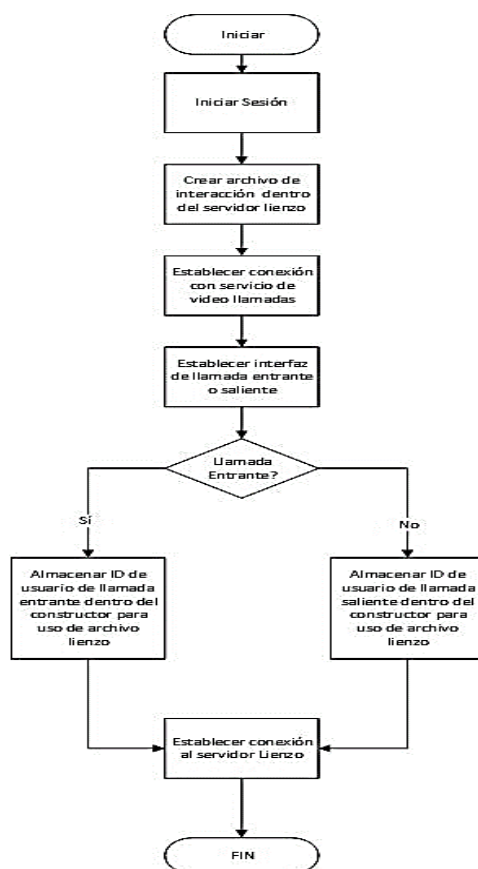


Figura 5. Diagrama de flujo usado en la lógica de comunicación mediante la interfaz de videollamada.

El principio de esta solución consiste en usar un archivo plano creado automáticamente al iniciar sesión con un identificador único por cada usuario. Este “ID” es el correo electrónico que también se utiliza dentro de la interfaz de videollamada. Mediante el uso de estas interfaces se obtiene el identificador del usuario que realiza la llamada y se lo asigna a un constructor.

El constructor permite comunicar a cada clase que interviene en la interacción del lienzo que archivo en específico se debe utilizar para establecer la comunicación y la visualización de los trazos plasmados.

c. Búsqueda de usuarios mediante el nombre o correo utilizando la instrucción UNION ALL

Al simplificar las búsquedas de los usuarios y contactos mediante una nueva estructura de consulta, permite al usuario la facilidad de encontrar a un contacto en específico mediante el correo electrónico y/o por coincidencias de nombres.

El uso de la sintaxis “UNION ALL” permite mejorar la versatilidad de las consultas a la base de datos, al evitar el uso de la sintaxis “SELECT DISTINCT” la carga de trabajo dentro del servidor disminuye y se obtiene mejores resultados. Aunque existen datos duplicados en el resultado de la consulta, esto no genera aparentemente un problema para la aplicación móvil. Sin embargo, la optimización de la sentencia está abierta a evitar posibles fallos que se presenten en una futura actualización. [21] [30]

El código se presenta a continuación:

```

$consulta = "SELECT
`MAIL`,`NOMBRE`,`APELLIDO`,`ESTAD
O` FROM `usuario` WHERE `MAIL` IN
(SELECT      `MAIL_DOS`      FROM
`contacto` WHERE `MAIL` = ? and
`ESTADO` = 1) AND `MAIL` != ? AND
`NOMBRE` like (SELECT `NOMBRE`
FROM `usuario` WHERE `MAIL` = ?)
UNION      ALL      SELECT
`MAIL`,`NOMBRE`,`APELLIDO`,`ESTAD
O` FROM `usuario` WHERE `MAIL` IN
(SELECT      `MAIL_DOS`      FROM
`contacto` WHERE `MAIL` = ? and
`ESTADO` = 1) AND `MAIL` != ? AND
`NOMBRE` like ?";
  
```

5 Resultados

Las pruebas fueron realizadas a 20 personas con el fin de hacer uso y aprovechar las funcionalidades que ofrece la aplicación Pizarra Digital Interactiva.

En este apartado se puntualiza las pruebas realizadas tanto a la aplicación móvil, como a la aplicación servidor.

5.1 Pruebas

En la evaluación del segundo prototipo de la PDI se considera las características para el modelo de calidad de software (interna y externa), además de dos características del modelo de calidad de uso.

5.1.1 Calidad Interna y Externa

Dentro de este grupo se considera las siguientes características y subcaracterísticas para evaluar al segundo prototipo de la Pizarra Digital Interactiva:

a. Funcionalidad

Se evaluó la capacidad de la aplicación (móvil y servidor) para satisfacer las necesidades dentro del conjunto de las funcionalidades que lo conforman.

- Interoperabilidad: Se probó la aplicación móvil en versiones de Android desde la 6.0 hasta la 9.0.
- Adecuación: Se adaptó las primeras funcionalidades al segundo prototipo de la aplicación móvil y servidor, entre las principales se detalla en la descripción de la aplicación móvil en el apartado de reingeniería aplicado al primer prototipo de la pizarra digital interactiva.

b. Eficiencia

Se evaluó la capacidad de la aplicación (móvil y servidor) en el consumo de recursos al ofrecer prestaciones específicas.

- Cumplimiento de la eficiencia: Se evaluó a partir del uso de recursos utilizados en la transmisión de datos entre la aplicación móvil y servidor.
- Comportamiento con respecto al tiempo: Se midió tiempo de respuestas en base a pruebas de carga en funcionalidades específicas.

Para las pruebas que se realizaron internamente siguiendo las características de funcionalidad y eficiencia, se obtienen los siguientes resultados:

- Compatibilidad: La aplicación es compatible con el sistema Android 7 hasta la versión Android 9.0
- Tiempo de carga: El tiempo promedio de registro está dentro de un minuto, y el tiempo promedio para iniciar sesión está considerado dentro de los 30 segundos.

Además, se presenta una tabla comparativa de los tiempos de carga aplicados a cada servicio ofrecido por la aplicación servidor en su versión inicial y actual.

Tabla 1. Comparación de pruebas de carga realizadas a las versiones de la aplicación servidor

Descripción del servicio web al que se aplica la prueba de software	Aplicación Servidor V1.0		
	Número total de muestras	Tiempo de respuesta promedio (Segundos)	Porcentaje de error
Servicio de inicio de sesión	500	1,649	1,60%
Servicio de búsqueda contactos conocidos	500	1,484	0,86%
Servicio de búsqueda contactos desconocidos	500	1,358	1,20%
Servicio de escritura de archivo lienzo	500	1,972	1,31%
Servicio de lectura de archivo lienzo	500	2,036	1,97%
	Aplicación Servidor V2.0		
	Número total de muestras	Tiempo de respuesta promedio (Segundos)	Porcentaje de error
Servicio de inicio de sesión	500	0,841	0%
Servicio de búsqueda contactos conocidos	500	0,219	0%
Servicio de búsqueda contactos desconocidos	500	0,218	0%
Servicio de escritura de archivo lienzo	500	0,201	0%
Servicio de lectura de archivo lienzo	500	0,200	0%

- Privacidad: Los servicios asociados al uso de contraseñas utilizan el método de encriptación SHA1.

5.1.2 Modelo de Calidad de Uso

Dentro de este grupo se considera las siguientes características:

a. Efectividad

Se evalúa la aplicación móvil (Pizarra Digital Interactiva) brindando un trato directo al usuario.

b. Satisfacción

Se mide la actitud del usuario final al tener una experiencia directa con un producto de software.

Se miden estos parámetros con la siguiente valorización: Muy Bajo; Bajo; Medio; Alto; Muy Alto; considerando la escala del 1 al 5 respectivamente para cada ítem; Integradas con las siguientes preguntas puntuales:

- ¿Nivel de dificultad de la navegación en la aplicación?

- ¿Cuál es el nivel de dificultad para establecer comunicación con otro usuario?
- ¿Qué puntuación le pondría a la aplicación después de su experiencia?
- ¿La idea principal que deseaba transmitir se efectuó con éxito?
- ¿Las herramientas existentes ayudaron a transmitir su idea?
- ¿La velocidad de la aplicación es óptima para su uso personal?

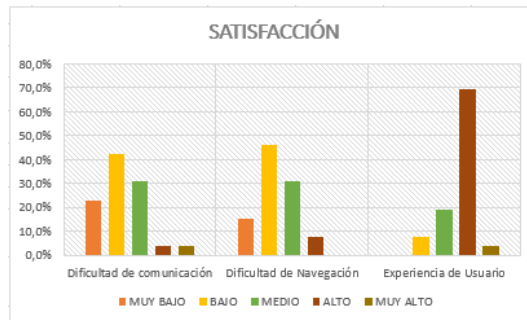


Figura 6. Resultados de las pruebas de Satisfacción

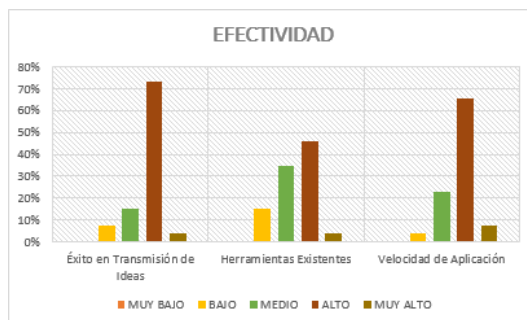


Figura 7. Resultados de las pruebas de Efectividad

5.2 Funcionalidades sugeridas y agregadas

Para elaborar un marco de posibles funcionalidades a agregarse en un futuro, se ha utilizado el método de Lean StartUp [31].

Este método define su proceso mediante el uso de un ciclo iterativo (Crear – Medir – Aprender), el cual permite elaborar prototipos minimalistas a partir de la retroalimentación obtenida mediante un grupo objetivo. En base a la retroalimentación aprendida a partir de las pruebas, se obtienen datos los cuales aportan al estudio y eventualmente a la creación de nuevas funcionalidades aplicables en versiones posteriores [32] [33].

A partir de las pruebas realizadas al segundo prototipo de la PDI, se enlistan las siguientes funcionalidades recomendadas:

Tabla 2. Funcionalidades sugeridas a implementarse en versiones posteriores.

Funcionalidades Sugeridas	Descripción	Nº sugerencias/usuario
Insertar figuras geométricas	Permitir crear figuras geométricas definidas.	4
Insertar imágenes	Permitir insertar imágenes almacenadas	6
Capas de dibujo	Gestionar varios lienzos al mismo tiempo	2
Insertar texto	Introducir texto en el lienzo	7
Grabar sesión de videollamada	Capturar video y audio de la sesión en el lienzo	2
Cambiar contraseña	Permitir actualizar la contraseña del usuario	8
Gestión de solicitudes de amigos	Aceptar y rechazar solicitudes de amistad	9

Dentro de las posibles funcionalidades sugeridas se ha agregado la funcionalidad de cambiar contraseña.

Se ha creado un servicio dentro del servidor para la funcionalidad que implementa seguridad mediante el método de encriptación SHA1.

6 Discusión

Las actividades de reingeniería de software presentaron resultados favorables para este caso de estudio, no obstante, el modelo fue ajustado a las necesidades que se presentaron a lo largo del análisis. Dentro de este proceso se visualizó una desactualización considerable dentro de los componentes los cuales afectaban directamente al funcionamiento de la aplicación del primer prototipo. Esto debido a que las tecnologías orientadas al desarrollo de software van quedando obsoletas en poco tiempo [34], aunque se recomienda utilizar las últimas tecnologías existentes, también se presenta un conflicto resultante de la incompatibilidad con algunos dispositivos.

Las optimizaciones fueron adaptadas con la idea de elevar la calidad y rendimiento que proporcionaba la aplicación, no obstante, parte de un extenso entendimiento de como la aplicación interactúa con otros servicios que actualmente son tecnologías más complejas que al inicio de su desarrollo. Esto compromete la implementación o actualización de alguno de sus servicios, pero también genera la necesidad que recaer en presentar un producto que ofrezca la funcionalidad y estabilidad que pretende transmitir al usuario final.

Se evidenció un mejor rendimiento al incluir actualizaciones en el servicio de video llamada, pero se debe considerar que al implementar una nueva arquitectura para el segundo prototipo la

carga sobre el servidor disminuyó, esto se debe a que, al no utilizar la misma cantidad de recursos y dividir los servicios, los resultados obtenidos con la arquitectura inicial del primer prototipo no fueron los más favorables.

Esto señala que las actualizaciones ayudan significativamente al desempeño de los productos de software, aunque en continuidad, limitan la funcionalidad en versiones anteriores de la plataforma en donde se ejecutan o para las que fueron diseñadas inicialmente.

En la actualidad, el usuario demanda mayor eficiencia en el uso de aplicaciones móviles, esta es una realidad latente para los programadores y la industria de desarrollo software. Al considerar al usuario como pilar fundamental para concebir nuevos requerimientos, se genera la necesidad de mejorar una aplicación existente o, crear un nuevo producto con mejores prestaciones logrando así, satisfacer a cabalidad las necesidades del usuario. Posterior a todo este análisis se puede reflexionar entonces: ¿Es el usuario quien demanda nuevas funcionalidades y esto conlleva a la abrumadora evolución tecnológica o la industria de HW y SW son quienes crean dicha tecnología para establecer una necesidad en el usuario?

7 Conclusiones

El estudio realizado en este proyecto demostró que el soporte y mantenimiento proporcionado a un producto de software, es directamente proporcional al desempeño del mismo, esto implica que la falta de este servicio puede desencadenar en la ineficiencia del software y eventualmente en su obsolescencia.

El método de reingeniería a través de su proceso presentó la oportunidad de profundizar en el entendimiento del primer prototipo de la aplicación Pizarra Digital Interactiva y con ello, exponer un escenario favorable para mejorar su estructura y funcionalidad dando como resultado un segundo prototipo, empleando tecnologías actuales, las cuales otorgan mayor fiabilidad y experiencia hacia el usuario final.

Los resultados expuestos por el ambiente de pruebas aplicados al segundo prototipo de Pizarra Digital Interactiva, demostraron que la implementación de una nueva arquitectura, mejora notablemente la eficiencia del prototipo, disminuyendo los tiempos de respuesta y el consumo de recursos dentro del servidor.

En versiones posteriores al segundo prototipo o en un ambiente de producción, el mantenimiento respecto a las actualizaciones y versiones tanto de la aplicación móvil como el

servidor, deben considerar una estrategia clara para mitigar los problemas expuestos en el primer prototipo de la Pizarra Digital Interactiva.

La transmisión de datos con respecto a los trazos en el lienzo aumenta considerablemente con resoluciones altas presentes en los dispositivos móviles actuales, las cuales afectan a la fluidez con la que se visualizan los trazos, por lo que implementar un servicio dedicado para solventar este problema es fundamental para disminuir el uso de recursos, elevando la calidad de dibujo y mejorando la experiencia que se entrega al usuario final.

El proceso llevado a cabo en la obtención de nuevas funcionalidades para la PDI, demostró que las necesidades del usuario tienden a incrementar, esto es parte de la evolución tecnológica que pretende mejorar la experiencia que se entrega al usuario final, por lo que el cambio y perfeccionamiento de la PDI está abierta a continuar en futuros trabajos.

Referencias

- [1] Statista Inc, «The Statistics Portal,» Statista Inc, Marzo 2019. [En línea]. Available: <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>. [Último acceso: 14 Mayo 2019].
- [2] Centro de Ciberseguridad Industrial, «Cómo impacta en el negocio el uso de software sin soporte Ejemplo Windows XP en el mundo industrial,» Centro de Ciberseguridad Industrial, Madrid, 2013.
- [3] V. A. Herrera Caldas, *DESARROLLO DE UN PLAN DE GESTIÓN DE MANTENIMIENTO DE SOFTWARE PARA EL DEPARTAMENTO DE SISTEMAS DE LA UNIVERSIDAD POLITÉCNICA SALESIANA BASADO EN LA NORMA ISO/IEC 14764:2006*, Cuenca: UNIVERSIDAD POLITÉCNICA SALESIANA, 2015.
- [4] C. F. Lincango Guamán y J. A. Villa Morales, *Desarrollo de un prototipo de pizarra digital interactiva para dispositivos móviles con sistema operativo Android, que incorpore video conferencia: Aplicación móvil.*, Quito: Universidad Politécnica Salesiana, 2017.
- [5] D. E. Galarza García, *Desarrollo de un prototipo de pizarra digital interactiva para dispositivos móviles con sistema operativo Android, que incorpore video*

- conferencia: *Aplicación Servidor.*, Quito: Universidad Politécnica Salesian, 2017.
- [6] R. Pressman, «Reingeniería,» de *Ingeniería de Software Un enfoque practico*, México, The McGraw-Hill Companies, Inc, 2010, pp. 658-673.
- [7] I. Sommerville, «Reingeniería,» de *Ingeniería de Software*, México, Pearson, 2011, pp. 248-261.
- [8] M. CALLEJAS-CUERVO, A. C. ALARCÓN-ALDANA y A. M. ÁLVAREZ-CARREÑO, «Modelos de calidad del software, un estado del arte,» *Entramado*, vol. 13, nº 1, pp. 236-250, 2017.
- [9] A. Abran, A. Khelifi, W. Suryn y A. Seffah, «Usability Meanings and Interpretations in ISO,» *Software Quality Journal*, vol. XI, pp. 325-338, 2003.
- [10] S. Herrera, P. Najar Ruiz, S. Rocabado, M. Fennema y M. Cianferoni, «Optimización de la Calidad de los Sistemas Móviles,» de *XV Workshop de Investigadores en Ciencias de la Computación*, Buenos Aires, 2013.
- [11] M. A. Abud Figueroa, «Calidad en la Industria del Software. La Norma ISO-9126,» 04 01 2004. [En línea]. Available: <https://www.repositoriodigital.ipn.mx/bitstream/123456789/5321/2/34-2.pdf>. [Último acceso: 22 05 2019].
- [12] A. M. García Rodríguez, Y. T. Casañola y A. Perdomo Vergara, «Optimización de estados en la mejora de procesos de software,» *Revista Venezolana de Información*, nº 2, pp. 9-27, 2013.
- [13] K. Sierra y B. Bates, *Praise for Head First Java*, United States of America: O'Reilly Media, 2005.
- [14] Android Developers, «Android Developers,» Google, 22 07 2018. [En línea]. Available: <https://developer.android.com/studio/intro>. [Último acceso: 15 04 2019].
- [15] D. García González, *Análisis, diseño e implementación de un SDK híbrido basado en tecnologías web para la integración de clientes contra una plataforma de servicios RESTful/JSON*, Madrid: E.T.S.I. de Sistemas Informáticos (UPM), 2015.
- [16] Oracle Corporation, «MySQL,» Oracle Corporation, 2019. [En línea]. Available: <https://www.mysql.com/>. [Último acceso: 20 05 2019].
- [17] R. Juárez-Ramírez, G. Licea y A. Cristóbal-Salas, «Ingeniería Inversa y Reingeniería Aplicadas a Proyectos de Software Desarrollados,» *Revista Iberoamericana de Sistemas, Cibernética e Informática*, vol. 4, nº 2, pp. 48-53, 2007.
- [18] AB, Sinch, «Sinch Video,» Sinch AB, 2019. [En línea]. Available: <https://www.sinch.com/docs/video/index.html>. [Último acceso: 16 04 2019].
- [19] J. C. Álvarez García, M. Mateos-Sánchez y M. Moreno García, «Metodología de reingeniería del software para la remodelación de aplicaciones científicas heredadas,» Departamento de Informática y Automática - Universidad de Salamanca, Salamanca, 2005.
- [20] PHP Group, «PHP,» 2019. [En línea]. Available: <https://www.php.net/manual/es/migration70.php>. [Último acceso: 01 05 2019].
- [21] Oracle Corporation, «MySQL 8.0 Reference Manual,» 2019. [En línea]. Available: <https://dev.mysql.com/doc/refman/8.0/en/>. [Último acceso: 08 05 2019].
- [22] L. Debrauwer y Y. Evain, «Patron Adapter,» de *PATRONES DE DISEÑO EN PHP: LOS 23 MODELOS DE DISEÑO: DESCRIPCIONES Y SOLUCIONES ILUSTRADAS EN UML2 Y PHP*, Barcelona, Ediciones ENI, 2015, pp. 96-106.
- [23] Android Developer, «Developer Android - Support library,» 2019. [En línea]. Available: <https://developer.android.com/topic/libraries/support-library>. [Último acceso: 02 05 2019].
- [24] M. Garriga, *Selección, Prueba y Adaptación de Servicios para Integración en Aplicaciones Orientadas a Servicios*, Buenos Aires: Repositorio Institucional Digital de Acceso Abierto, 2015.
- [25] J. Ferrer Martínez, *Implantación de aplicaciones web (GRADO SUP.)*, Madrid: RA-MA, 2012.
- [26] AB, Sinch, «Video for Android,» Sinch AB, 2019. [En línea]. Available: <https://www.sinch.com/docs/video/android/index.html>. [Último acceso: 03 05 2019].
- [27] J. E. Amaro Soriano, *El gran libro de programación avanzada con Android*, Mexico: Alfaomega Grupo Editor, 2012.
- [28] H. S. Oluwatosin, «Client-Server Model,» *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 16, nº 1, pp. 67-71, 2014.
- [29] L.-Å. Fredlund, C. Benac Earle, Á. Herranz y J. Mariño, «Property-Based Testing of JSON Based Web Services,» de

2014 IEEE International Conference on Web Services, Anchorage, 2014.

- [30] I. GILFILLAN, «Combinaciones Internas,» de *LA BIBLIA DE MYSQL*, México, ANAYA MULTIMEDIA, 2003, pp. 140-148.
- [31] L. Kullmar y I. Lallerstedt, *The Lean Startup Method*, Lund: Department of Design Sciences, Faculty of Engineering LTH, Lund University, 2017.
- [32] E. Ries, *El Método LEAN STARTUP*, Barcelona: Grupo Planeta,, 2012.
- [33] F. J. Llamas Fernández y J. C. Fernández Rodríguez, «La metodología Lean Startup: desarrollo y aplicación para el emprendimiento,» *Revista EAN*, n° 84, pp. 79-95, 2018.
- [34] M. Ortiz Herrera, *MÉTODOS Y TÉCNICAS PARA LA GESTIÓN DE PROYECTOS SOFTWARE*, Sevilla: DEPARTAMENTO DE ORGANIZACIÓN INDUSTRIAL Y GESTIÓN DE EMPRESAS, 2010.